ROBERT A. CESARI (1928-2008)
JOHN F. MCKENNA
MARTIN J. O'DONNELL
THOMAS C. O'KONSKI
PATRICIA A. SHEEHAN
MICHAEL E. ATTAYA
CHARLES J. BARBAS
MICHAEL R. REINEMANN
KEVIN GANNON
DIANE H. DIEGER
JAMES A. BLANCHETTE
JAMES M. BEHMKE
SHANNEN C. DELANEY
OMAR M. WADHWA
RITA M. ROONEY
MICHAEL T. ABRAMSON
STEPHEN D. LEBARRON

CESARI AND MCKENNA, LLP
ATTORNEYS AT LAW
88 BLACK FALCON AVENUE
BOSTON, MASSACHUSETTS

Telephone: (617) 951-2500 Telecopier: (617) 951-3927
Website: www.c-m.com

INTELLECTUAL PROPERTY AND RELATED CAUSES

A. SIDNEY JOHNSTON
EDWIN H. PAUL
OF COUNSEL

HEATHER SHAPIRO
PATENT AGENT

FACSIMILE COVER SHEET

112056-0474

DATE:	February 16, 2010
TOTAL PAGES WITH COVER:	21
TO:	Giovanna B. Colan
FIRM:	United States Patent and Trademark Office
FACSIMILE NUMBER:	571-273-2752
TELEPHONE NUMBER:	571-272-2752
FROM:	Michael T. Abramson

COMMENTS:

Please call Kristin at 617-951-3089 to confirm receipt of this agenda.

Thank you!

SPECIAL INSTRUCTIONS:

If you do not receive all pages, or you are not the intended recipient, please contact us at (617) 951-2500 as soon as possible.

PATENTS
112056-0474
P01-2475.01

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re The Application of:
Hristo Iankov Bojinov

Serial No.: 10/803,788

Filed: March 17, 2004

For: METHOD AND APPARATUS FOR IMPROVING FILE SYSTEM PROXY PERFORMANCE AND SECURITY BY DISTRIBUTING INFORMATION TO CLIENTS VIA FILE HANDLES

Examiner: Colan, Giovanna B

Art Unit: 2162

Confirmation No.: 8050

Cesari and McKenna, LLP
88 Black Falcon Avenue
Boston, MA 02210
February 16, 2010

INTERVIEW AGENDA

(1) Explain the problem solved
(2) Analyze the claimed solution
(3) Analyze all cited art
(4) Explain why all claims are allowable in view of the cited prior art

Present Status of Case

This Agenda for a telephonic interview with Examiner Colan is sent in response to the FINAL Office Action mailed by USPTO on January 5, 2010.

- This Agenda for the telephonic interview is sent via facsimile:
  - **Facsimile** # (571)-273-2752
  - **Telephone** # (571)-272-2752
- At Examiner's earliest convenience, please call Attorney Michael T. Abramson (Reg. No. 60,320) at 617-951-3053 to schedule the telephonic interview. Thank you in advance for your time.

IN THE CLAIMS:

1. (Previously Presented) A method for establishing identity in a file system, comprising:

receiving, from a client, a first Network File System (NFS) operation concerning an indicated file, the first NFS operation received by a proxy;

forwarding the first NFS operation from the proxy to be received by a file server;

returning a NFS file handle associated with the first NFS operation from the file server to the proxy in response to the file server receiving the first NFS operation from the proxy;

inserting, by the proxy, metadata into the NFS file handle in response to receiving the NFS file handle from the file server, wherein the metadata is an encryption key;

sending, by the proxy in response to receiving the NFS file handle from the file server, the NFS file handle with the metadata inserted in the NFS file handle to the client as a reply to the first NFS operation; and

using, by the client, the metadata and the NFS file handle in a second NFS operation to identify the client and the indicated file.

2. (Previously Presented) The method of Claim 1, whereby using the metadata in the NFS file handle eliminates a need for the proxy to generate additional requests to the file server to establish file identity, and for completing client requests.

3. (Previously Presented) The method of Claim 1, further comprising:

encoding metadata in a form of a session key into the file handle, the session key expiring after a predetermined amount of time.

4. (Previously Presented) The method of Claim 1, further comprising:

using an NFS file system as the file system.

5. (Previously Presented) The method of Claim 1, further comprising:

using a stateless protocol by the file system.

6-29. (Cancelled).

30. (Previously Presented) The method of claim 1, further comprising:

receiving, from the client, a second NFS operation by the proxy, the second NFS operation comprising the metadata in a further NFS file handle sent with the second NFS operation;

identifying, in response to the metadata, the client as having a permission to submit the second NFS operation;

sending the second NFS operation to the file server and not sending the metadata with the second NFS file handle to the file server; and

receiving by the proxy a further NFS reply from the file server, and sending by the proxy the further NFS reply to the client.

31. (Previously Presented) A method for establishing identity in a file system, comprising:

receiving a first file request concerning an indicated file from a client, the first file request received by a proxy;

forwarding the first file request from the proxy to a file server;

returning a reply associated with the first file request from the file server to the proxy, wherein the reply includes a file handle associated with the indicated file;

inserting, by the proxy, metadata into the file handle;

sending, by the proxy, the file handle with the metadata inserted in the file handle to the client, the metadata to be used in further requests to identify the client as having a permission to access the indicated file;

receiving, from the client, a second file request by the proxy, the second file request including the metadata in a second file handle sent with the second file request;

identifying, in response to the metadata, that the client has the permission to submit the second file request;

sending the second file request to the file server and not sending the metadata with the second file handle to the file server; and

receiving by the proxy a second reply from the file server, and sending by the proxy the second reply to the client.

32. (Previously Presented) An apparatus to establish identity in a file system, comprising:

a proxy configured to receive a first Network File System (NFS) operation concerning an indicated file sent by a client to the file system, the proxy further configured to forward the first NFS operation to be received by a file server;

the file server configured to return a NFS file handle associated with the first NFS operation to the proxy in response to the file server receiving the first NFS operation from the proxy;

the proxy further configured to insert metadata into the NFS file handle in response to receiving the NFS file handle from the file server, wherein the metadata is an encryption key; and

the proxy further configured to send the NFS file handle with the metadata inserted in the NFS file handle to the client as a reply to the first NFS operation, the metadata and the NFS file handle to be used in a second NFS operation to identify the client and the indicated file.

33. (Previously Presented) The apparatus as in claim 32, further comprising:

the proxy further configured to receive, by the client, a second NFS operation, the second NFS operation comprising the metadata in the second NFS file handle sent with the second NFS operation;

the proxy to identify, in response to the metadata, the client as having a permission to submit the second NFS operation;

the proxy to send the second NFS operation to the file server and not to send the metadata with the second NFS file handle to the file server; and

the proxy to receive a second NFS reply from the file server, and the proxy to send the second NFS reply to the client.

34. (Previously Presented) The apparatus of Claim 32, further comprising:
the proxy to use the metadata in the NFS file handle received from the client to eliminate a need for additional communication with the file server to establish file identity.

35. (Previously Presented) The apparatus of Claim 32, further comprising:
the proxy to encode the metadata in a form of a session key into the NFS file handle, the session key expiring after a predetermined amount of time.

36. (Previously Presented) The apparatus of Claim 32, further comprising:
an NFS file system used as the file system.

37. (Previously Presented) The apparatus of Claim 32, further comprising:
a stateless protocol used by the file system.

38. (Previously Presented) A non-volatile memory executed on a computer, comprising:
the non-volatile memory containing procedures for execution on the computer for a method of establishing identity in a file system, the method having the steps of,
receiving, from a client, a first Network File System (NFS) operation concerning an indicated file, the first NFS operation received by a proxy;
forwarding the first NFS operation from the proxy to be received by a file server;
returning a NFS file handle associated with the first NFS operation from the file server to the proxy in response to the file server receiving the first NFS operation from the proxy;
inserting, by the proxy, metadata into the NFS file handle in response to receiving the NFS file handle from the file server, wherein the metadata is an encryption key; and
sending, by the proxy in response to receiving the NFS file handle from the file server, the NFS file handle with the metadata inserted in the NFS file handle to the client as a reply to the first NFS operation; and

using, by the client, the metadata and the NFS file handle in a second NFS operation to identify the client and the indicated file.

39. (Previously Presented) A method for establishing identity in a file system, comprising:

receiving a first file request concerning an indicated file from a client, the first file request received by a proxy;

forwarding the first file request from the proxy to a file server;

granting a permission for the request to be acted upon by the file system in response to a predetermined protocol;

returning a reply associated with the first file request from the file server to the proxy, wherein the reply includes a file handle associated with the indicated file;

inserting, by the proxy, a session key into the file handle; and

sending, by the proxy, the file handle with the session key inserted in the file handle to the client, the session key to be used in further requests to identify the client and the indicated file.

40. (Previously Presented) The non-volatile memory of Claim 38, further comprising:

receiving, from the client, a second NFS operation by the proxy, the second NFS operation comprising a session key in a second NFS file handle sent with the second NFS operation;

identifying, in response to the session key, that the client has the permission to submit the second NFS operation;

sending the second NFS operation to the file server and not sending the session key with the second NFS file handle to the file server; and

receiving by the proxy a second NFS reply from the file server, and sending by the proxy the second NFS reply to the client.

41. (Previously Presented) The non-volatile memory of Claim 40, further comprising:

causing the session key to expire after a selected amount of time.

42. (Previously Presented) The non-volatile memory of Claim 40, further comprising:
causing the session key to expire after a selected amount of usage.

43. (Previously Presented) The non-volatile memory of Claim 38, further comprising:
using a NFS file server as the file server.

44. (Previously Presented) The non-volatile memory of Claim 38, further comprising:
using a two way communication exchange between the proxy and the file server.

45. (Previously Presented) An apparatus to establish identity in a file system, comprising:
a proxy to receive a file request sent by a client to the file system, the proxy to forward the request to a file server;
the file server to return a reply associated with the file request to the proxy, wherein the reply includes a file handle;
the proxy to insert a session key into the file handle; and
the proxy to send the file handle with the session key inserted in the file handle to the client, the session key to be used in further requests to identify the client and the indicated file.

46. (Previously Presented) The apparatus as in claim 45, further comprising:
the proxy to receive, by the client, a second file request, the second file request to include the session key in a further file handle sent with the second request;
the proxy to identify, in response to the session key, the client as having a permission to submit the another file request;
the proxy to send the second request to the file server and not to send the session key with the second file handle to the file server; and
the proxy to receive a further reply from the file server, and the proxy to send the further reply to the client.

47. (Previously Presented) The apparatus of Claim 45, further comprising:
the proxy to use the metadata in the file handle received from the client to eliminate a need for additional communication with the file server to establish file identity.

48. (Previously Presented) The apparatus of Claim 45, further comprising:
the proxy to encode the metadata in a form of a session key into the file handle, the session key expiring after a predetermined amount of time.

49. (Previously Presented) The apparatus of Claim 45, further comprising:
an NFS file system used as the file system.

50. (Previously Presented) The apparatus of Claim 45, further comprising:
a stateless protocol used by the file system.

51. (Previously Presented) An apparatus to establish identity in a file system, comprising:
a proxy configured to receive a first file request sent by a client to the file system, the proxy further configured to forward the first file request to a file server;
the file server configured to return a reply associated with the first file request to the proxy;
the proxy further configured to insert a session key into a file handle;
the proxy further configured to send the file handle with the session key inserted in the file handle to the client, the session key configured to be used in a second file request to identify the client and the indicated file;
the proxy further configured to receive, by the client, a second file request, the second file request configured to include the session key in a second file handle sent with the second file request;
the proxy further configured to identify, in response to the session key, the client as having a permission to submit the second file request;

the proxy further configured to send the second file request to the file server and not to send the session key with the second file handle to the file server; and

the proxy further configured to receive a second reply from the file server, and the proxy further configured to send the second reply to the client.

52. (Previously Presented) A method for establishing identity in a file system, comprising:

receiving a first file request concerning an indicated file from a client, the first file request received by a proxy;

forwarding the first file request from the proxy to a file server;

determining that the client has a permission to have the request acted upon by the file system in response to a predetermined protocol;

returning a reply associated with the first file request from the file server to the proxy, wherein the reply includes a file handle associated with the indicated file;

inserting, by the proxy, a cryptographic information into the file handle;

sending, by the proxy, the file handle with the cryptographic information inserted in the file handle to the client, the cryptographic information to be used in one or more requests to identify the client and the indicated file.

53. (Previously Presented) The method according to claim 52, further comprising:

receiving, by the client, a second file request by the proxy, the second file request including the cryptographic information in a second file handle sent with the second file request;

identifying, in response to the cryptographic information, that the client has the permission to submit the second file request;

sending the second file request to the file server and not sending the cryptographic information with the second file handle to the file server; and

receiving by the proxy a second reply from the file server, and sending by the proxy the second reply to the client.

54. (Previously Presented) The method according to claim 52, further comprising:
causing the cryptographic information to expire after a selected amount of time.

55. (Previously Presented) The method according to claim 52, further comprising:
causing the cryptographic information to expire after a selected amount of usage.

56. (Previously Presented) The method according to claim 52, further comprising:
using a NFS protocol as the predetermined protocol.

57. (Previously Presented) The method according to claim 52, further comprising:
using as the predetermined protocol a two way communication exchange between the proxy and the file server.

58. (Previously Presented) An apparatus to establish identity in a file system, comprising:
a proxy configured to receive a file request for an indicated file sent by a client to the file system, the proxy further configured to forward the request to a file server;
the file server configured to return a reply associated with the file request to the proxy, wherein the reply is configured to include a file handle;
the proxy further configured to insert a cryptographic information into the file handle; and
the proxy further configured to send the file handle with the cryptographic information inserted in the file handle to the client, the cryptographic information configured to be used in further requests to identify the client and the indicated file.

59. (Previously Presented) The apparatus as in claim 58, further comprising:
the proxy further configured to receive, by the client, a second request, the second file request to include the cryptographic information in a second file handle sent with the second request;

the proxy further configured to identify, in response to the cryptographic information, the client as having a permission to submit the second file request;

the proxy further configured to send the second request to the file server and not to send the cryptographic information with the second file handle to the file server; and

the proxy further configured to receive a further reply from the file server, and the proxy to send the further reply to the client.

60. (Previously Presented) The apparatus of claim 58, further comprising:

the proxy further configured to use the metadata in the file handle received from the client to eliminate a need for additional communication with the file server to establish file identity.

61. (Previously Presented) The apparatus of claim 58, further comprising:

the proxy further configured to encode the metadata in a form of a cryptographic information into the file handle, the cryptographic information configured to expire after a predetermined amount of time.

62. (Previously Presented) The apparatus of claim 58, further comprising:

an NFS file system used as the file system.

63. (Previously Presented) The apparatus of claim 58, further comprising:

a stateless protocol used by the file system.

64. (Previously Presented) An apparatus to establish identity in a file system, comprising:

a proxy configured to receive a first file request sent by a client to the file system, the proxy to forward the first file request to a file server;

the file server configured to return a reply associated with the first file request to the proxy;

the proxy further configured to insert a cryptographic information into a file handle;

the proxy further configured to send the file handle with the cryptographic information inserted in the file handle to the client, the cryptographic information configured to be used in a second file request to identify the client and the indicated file;

the proxy further configured to receive, by the client, a second file request, the second file request configured to include the cryptographic information in a second file handle sent with the second file request;

the proxy further configured to identify, in response to the cryptographic information, the client as having a permission to submit the second file request;

the proxy further configured to send the second file request to the file server and not to send the cryptographic information with the second file handle to the file server; and

the proxy further configured to receive a second reply from the file server, and the proxy to send the second reply to the client.

65. (Previously Presented) A method for establishing identity in a file system, comprising:

receiving a file request concerning an indicated file from a client, the request received by a proxy;

forwarding the request from the proxy to a file server;

returning a reply associated with the file request from the file server to the proxy, wherein the reply includes a file handle associated with the indicated file;

inserting, by the proxy, metadata into the file handle; and

sending, by the proxy, the file handle with the metadata inserted in the file handle to the client, a size of the file handle set to a sum of a length of the server file handle and a length of the proxy metadata, the metadata to be used in further requests to identify the client and the indicated file.

66. (Previously Presented) A method, comprising:

receiving, by a proxy, a file request for a file sent from a client;

forwarding the file request from the proxy to a file server;

returning a reply associated with the file request from the file server to the proxy, wherein the reply includes a file handle;

inserting, by the proxy, metadata into the file handle;

sending, by the proxy, the file handle with the metadata inserted in the file handle to the client; and

using, by the client, the metadata inserted into the file handle in a subsequent file request to identify the client and the file.

67. (Previously Presented) A computer apparatus, comprising:

a proxy configured to receive a client file request for a file and forward the file request from the proxy to a file server;

the server configured to return a reply associated with the file request, wherein the reply includes a file handle;

the proxy further configured to intercept the file handle sent from the server and insert metadata into the file handle to create a modified file handle;

the proxy further configured to send the modified file handle with the metadata inserted in the file handle to the client; and

the proxy further configured to receive the modified file handle from the client for a second file request for the file, wherein the proxy is further configured to use the modified file handle to eliminate a need for the proxy to generate one or more additional requests to the server that would be required to access the file if the modified file handle did not include the inserted metadata.

PATENTS
112056-0474
P01-2475.01

REMARKS

Claims 1-5 and 30-67 are in the case.

Claim 1 is presented for discussion.

Rejections Under 35 U.S.C. §103

At Paragraph 6 of the Office Action, claims 1-5 and 30-67 were rejected under 35 U.S.C. §103(a) as being obvious over Chandrashekhar et al., U. S. Patent Publication 2005/0033988 published on February 10, 2005 (hereinafter "Chandrashekhar"), and in view of Ryuutou et al., U.S. Patent Application Publication No. 2002/0083191 published on June 27, 2002 (hereinafter "Ryuutou").

Applicant's claimed novel and non-obvious invention, as set out in representative claim 1, comprises in part:

> 1. A method for establishing identity in a file system, comprising:
> receiving, from a client, a first Network File System (NFS) operation concerning an indicated file, the first NFS operation received by a proxy;
> forwarding the first NFS operation from the proxy to be received by a file server;
> returning a NFS file handle associated with the first NFS operation from the file server to the proxy in response to the file server receiving the first NFS operation from the proxy;
> inserting, by the proxy, metadata into the NFS file handle in response to receiving the NFS file handle from the file server, wherein the metadata is an encryption key;
> **sending**, by the proxy in response to receiving the NFS file handle from the file server, **the NFS file handle *with the metadata inserted in the NFS file handle to the client*** as a reply to the first NFS operation; and
> using, by the client, the metadata and the NFS file handle in a second NFS operation to identify the client and the indicated file.

As an aside, in an interview conducted on August 27, 2009 and in the subsequent Amendment filed on September 8, 2009, Applicant discussed how neither Chandrashekhar nor Ryuutou disclosed an **NFS file handle**. As such, Applicant argued that neither Chandrashekhar nor Ryuutou failed to teach or suggest Applicant's claimed

novel and non-obvious ***inserting encryption key metadata into an NFS file handle*** and then **sending the NFS file handle** ***with the encryption key metadata inserted in the NFS file handle to the client as a reply to the first NFS operation.*** Applicant maintains this argument; however, in the interest of advancing prosecution and avoiding the delay in seeking appellate review from the Board of Patent Appeals and Interferences and/or the U.S. Court of Appeals for the Federal Circuit, Applicant respectfully presents an alternative argument. Applicant expressly reserves the right to present these contentions or variations thereof in any appellate procedures.

Chandrashekhar discusses processing file requests sent by a client and received by a proxy using security applications to encrypt, decompress, verify, and decrypt network data by a server receiving the files from the proxy [0058; 0071]. Header policy information is determined, generated, and then stored on the filer server [0055; Fig. 4-5]. However, any metadata added to a file is stripped off before the file is returned to the client [0038].

Ryuutou discloses, in relevant part as cited by Examiner, a client establishing an HTTP connection between the client and a proxy server by initiating a communication connection request [0072-0073]. A session ID is added to header information of an HTTP request [0067; *see also* Fig. 9 below]. Notably, Ryuutou explicitly states that session IDs are identifiers of sessions whose communications have been established [0069; *see also* Fig. 1 below (item 2: "DETERMINING WHETHER OR NOT CONNECTION IS ESTABLISHED ACCORDING TO IDENTIFIER WRITTEN IN REQUEST")].

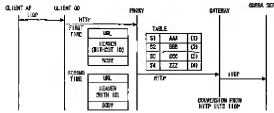

FIG. 9

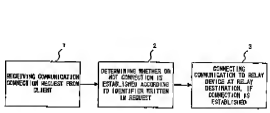

FIG. 1

Applicant respectfully urges that Chandrashekhar, taken singly or in any combination with Ryuutou, does not disclose Applicant's claimed novel and non-obvious use of

sending a NFS file handle ***with encryption key metadata inserted in the NFS file handle to a client.***

Applicant claims, in part, a proxy receiving from a client a first Network File System (NFS) operation concerning an indicated file and forwarding the first NFS operation from the proxy to be received by a file server. Applicant further claims returning a **NFS file handle** associated with the first NFS operation from the file server to the proxy in response to the file server receiving the first NFS operation from the proxy. Applicant further claims inserting, by the proxy, ***encryption key metadata*** into the **NFS file handle**. With that being said, after inserting (the encryption key) metadata into the NFS file handle, Applicant further claims **sending a NFS file handle** ***with encryption key metadata inserted in the NFS file handle to a client.***

Applicant respectfully argues that Chandrashekhar does not teach or suggest Applicant's claimed novel and non-obvious **sending a NFS file handle** ***with encryption key metadata inserted in the NFS file handle to a client.*** Specifically, while Chandrashekhar may or may not disclose inserting encryption key metadata into a NFS file handle, Chandrashekhar is explicit in stating that the metadata is stripped off ***before*** the file is returned to the client (see Chandrashekhar at [0038] cited, in relevant part, below):

> …The meta-data is stripped off ***before*** the file data/file attributes are returned to the client… (emphasis added)

Accordingly, not only does Chandrashekhar not teach Applicant's claimed novel and non-obvious **sending a NFS file handle** ***with encryption key metadata inserted in the NFS file handle to the client,*** but Chandrashekhar actually teaches away from doing so. Therefore, because Chandrashekhar explicitly teaches away from **sending a NFS file**

handle ***with encryption key metadata inserted in the NFS file handle to the client,*** Chandrashekhar fails to teach or suggest Applicant's claimed novel **sending a NFS file handle** ***with encryption key metadata inserted in the NFS file handle to the client.***

Applicant respectfully argues that Ryuutou does not teach or suggest Applicant's claimed novel and non-obvious **sending a NFS file handle** ***with encryption key metadata inserted in the NFS file handle to a client.*** Specifically, while Ryuutou may or may not teach adding a session ID to header information, Ryuutou explicitly states that a session ID is an identifier of a session between a client and a server whose communications have been started (see Ryuutou at [0069] cited, in relevant part, below):

> The already stored session identifiers…are identifiers of sessions whose communications have been started…(emphasis added)

In other words, while Ryuutou may or may not teach adding a session ID to header information, Ryuutou's session ID is not ***encryption key metadata.*** In contrast, Applicant claims **sending** ***encryption key metadata inserted in the NFS file handle to the client.*** As such, because Ryuutou's definition of a session ID does not include any disclosure of being encryption key metadata, Ryuutou fails to teach or suggest Applicant's claimed novel **sending a NFS file handle** ***with encryption key metadata inserted in the NFS file handle to the client.***

To reiterate:

(I) Even if it is assumed *arguendo* that Chandrashekhar stores encryption key metadata in a NFS file handle, Chandrashekhar does not **send the NFS file handle** ***with encryption key metadata inserted in the NFS file handle to the client***, because Chandrashekhar explicitly states that the metadata is stripped off ***before*** the file is returned to the client. Thus, if encryption key metadata is stored in a NFS file handle, but stripped off before the file handle is returned to the client, then the file handle cannot contain encryption key metadata. Therefore, Chandrashekhar fails to teach or suggest

Applicant's claimed novel and non-obvious **sending a NFS file handle** ***with encryption key metadata inserted in the NFS file handle to the client***.

(II) Similarly, even if it is assumed *arguendo* that Ryuutou stores a session ID into a NFS file handle, Ryuutou does not **send the NFS file handle** ***with encryption key metadata inserted in the NFS file handle to the client***, because Ryuutou explicitly states that a session ID is an identifier of a session between a client and a server whose communications have been started. Thus, if a session ID is an identifier of a session between a client and a server whose communications have been started, then a session ID cannot be encryption key metadata. More specifically, if a session ID cannot be encryption key metadata, then sending a NFS file handle with a session ID inserted in the NFS file handle to the client is demonstrably different than **sending a NFS file handle** ***with encryption key metadata inserted in the NFS file handle to the client***. Therefore, Ryuutou fails to teach or suggest Applicant's claimed novel and non-obvious **sending a NFS file handle** ***with encryption key metadata inserted in the NFS file handle to the client***.

Accordingly, Applicant respectfully urges that Chandrashekhar, taken singly or in any combination with Ryuutou, is legally insufficient to render the presently claimed invention obvious under 35 U.S.C. §103. Chandrashekhar and Ryuutou, taken singly or in any combination, fails to teach or suggest Applicant's claimed novel and non-obvious **sending a NFS file handle** ***with encryption key metadata inserted in the NFS file handle to a client***.

Applicant's Interpretation of the Prior Art

Applicant's interpretation of the prior art references was derived, in part, from the following excerpts:

Chandrashekhar

> [0038]…The meta-data relates to key management, length of the original file/dataset, whether the file was compressed prior to encryption or not,

integrity checks for file data. The meta-data is stripped off ***before*** the file data/file attributes are returned to the client… (emphasis added)

Ryuutou

[0017] …a communication connection request is received from a client, whether or not a communication connection corresponding to a series of communications is established is determined according to an identifier written in the communication connection request, and the requested communication is connected to a particular relay device as a relay destination of an established communication connection, if the communication connection is established. (emphasis added)

[0057] FIG. 6 is a flowchart showing the process of a communication connection management method in this preferred embodiment. In FIG. 5, when a new communication connection to a gateway is established in correspondence with the initial communication connection request within one session, and a session ID is set, its contents are stored in a memory (table) not shown...(emphasis added)

[0069] The already stored session identifiers and connection numbers are identifiers of sessions whose communications have been started, and numbers of connections established for the sessions respectively. (emphasis added)

[0072] As explained with reference to FIG. 9, the session number S4, and the session ID ZZZ are set by the proxy in correspondence with this communication connection request. The newly set session ID is added to the header information…and returned…to the client side. (emphasis added)

[0073] At this time, ZZZ as the session ID is added between B and C in the header information shown in FIG. 10. As a method adding a session ID, a method such as Netscape Cookie, with which a browser side can recognize and store, for example, data that is additionally described in an HTTP header, is used. (emphasis added)

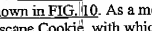

FIG. 10

[0074] A reply including header information to which a session ID is

added is returned from a proxy side to a PC side as described above, so that header information including the session ID can be used as the header information in the second and subsequent communication connection requests. (emphasis added)

Conclusion

All newly proposed claims and/or proposed claim amendments are believed to be fully supported by Applicant's specification.

All proposed independent claims are believed to be in condition for allowance.

All proposed dependent claims are believed to be dependent from allowable proposed independent claims, and therefore in condition for allowance.